Databáze bez databázového programu

Petr Olšák

Před nedávnem jsem se nabídl Karlovi Horákovi, že budu spravovat databázi členů našeho sdružení. Po jisté „jednací době“ jsem od něj obdržel příslušnou databázi ve formátu DBase. Ukázalo se, že struktura této databáze není příliš dobře dořešena a že bude potřeba změnit a doplnit plno údajů.

Sám jsem nechtěl mít nic společného s žádným databázovým systémem. Po rozvaze, co budu od databáze očekávat, jsem dospěl k jednoznačnému závěru, že na doplňování a změny údajů vystačím s obyčejným editorem, rešerše a tisky mi udělá TEX a řazení podle české abecedy program `csr`. Řazení podle jednodušších položek, (např. podle směrovacího čísla) zvládne funkce „Sort“ v editoru.

Databázi jsem transformoval z vnitřní datové reprezentace příslušného databázového programu do textového souboru, kde jeden řádek odpovídal jednomu „formuláři“ a jednotlivé položky lícovaly pěkně pod sebou. Položky jsem dále oddělil celkem snadno pomocí blokových operací editoru jistými oddělovači (smluvenými znaky), aby je mohl správně rozlišit TEX. K manipulaci s databází jsem použil editor Qedit.

Neznám vymoženosti databázových programů. Už v jednom svém článku jsem přiznal, že databázové systémy vůbec neumím. Proto nemohu srovnávat výhody těchto systémů s navrženou koncepcí. Přesto se pokusím o uvedení několika důvodů, proč mi práce v editoru připadá přijatelnější.

- Není potřeba učit se ovládat nové uživatelské prostředí.
- S databází je možné bez problémů pracovat v UNIXu i v DOSu.
- Textový soubor mohu snadno posílat např. ostatním členům výboru a nenutit je při čtení údajů používat nějaký databázový systém. Když jim navíc pošlu makra pro zpracování TEXem, je vše zcela bez problémů.
- Je-li výjimečně v jednom formuláři nějaká položka delší (například výjimečně dlouhá e-mailovská adresa), nemusí se předělávat struktura databáze. Stačí dát této položce prostor na úkor sousední položky, protože při zpracování pro tisk mají stejně rozhodující roli oddělovací znaky a nikoli poloha údaje na řádku.

- Vyhledávání libovolného údaje probíhá snadno pomocí funkce „find“ v editoru.
- Prostým posunováním kurzoru ve vertikálním nebo horizontálním směru lze sledovat určitý údaj buď v kontextu s údaji stejného typu (např. platby za rok 1994) nebo v rámci jednoho řádku (jeden člen sdružení). Na obrazovce jsou vidět a snadno dosažitelné údaje v obou těchto směrech.
- Doplňování údajů (např. údaje o platbách členských příspěvků) lze provádět s minimálním počtem úhozů na klávesnici, protože pro poskočení kurzoru v řádku na požadované místo a pro doplnění stále se opakujících údajů (např. rok platby) lze udělat makro editoru.
- Kdykoli lze pomocí blokové operace třídění seřadit databázi tak, že je přehledně vidět sledovaný údaj. Například seřazení podle údaje „je člen výboru“ oddělí členy výboru od ostatních členů.

Přiznávám, že koncepce bude asi pro rozsáhlé databázové balíky údajů nedostačující. Napadají mě tyto nevýhody:

- Může se narazit na kapacitní možnosti editoru (např. pro Qedit je maximální délka řádku 512 znaků a maximální velikost souboru je omezena velikostí konvenční paměti DOSu).
- Může se jedinou neopatrnou operací v editoru poškodit struktura databáze tak, že náprava je těžko dosažitelná. Zvláště pravděpodobné to je v případě, kdy s databází pracují nepoučené osoby, které neumí správně zacházet s editorem.
- Je otázka, zda může posložit strukturovaný textový soubor jako vstup do databázového systému, ve kterém bude chtít pracovat můj následovník, který bude pokračovat v obhospodařování databáze členů našeho sdružení.

První nevýhoda nehrála v případě databáze členů $\mathcal{CS}$TUGu velkou roli, protože celá databáze má objem pouze 130 kB a délka jednoho řádku je zhruba 370 znaků. Ani druhou nevýhodu nepovažuji za důležitou, protože zálohovat jakoukoli práci je nutnost. Kdo nezálohuje, může na to tvrdě doplatit, i když používá kvalitní a „blbuvzdorný“ databázový systém. Třetí nevýhoda se asi vyskytuje obecně při přechodu od jednoho databázového systému k jinému. Pokud ovšem databázový systém neumí import dobře strukturovaných textových souborů, pak to je dosti špatný databázový systém.

Nyní se zastavím u dalších „součástí" této koncepce udržování databáze. Kromě editoru totiž pracuji ještě s dalšími dvěma softwarovými prostředky. Jedním z nich je program `csr` a druhým z nich je TEX. Oba programy jsou nezávislé na použitém systému (UNIX nebo DOS), a proto se takto koncipovaná databáze snadno udržuje na obou systémech zároveň.

Pokud jde o řazení podle abecedy, učinil jsem toto pozorování. Původní databáze, kterou jsem obdržel od Karla Horáka, měla snahu být seřazena podle jakési pseudoabecedy, kde všechny akcentované znaky byly později než Z. Že by existovalo něco jako dvojhláska CH, nebo dokonce víceprůchodový systém porovnávání, o tom nemoho být ani řeči. Podobně zmršené databázové výstupy dostávám každoročně jako seznamy studentů ze studijního oddělení. Dospěl jsem tedy (přiznám, že bez důkladnějšího průzkumu) k závěru, že existuje plno databázových systémů, které českou normu abecedního řazení slov neuznávají. Řekl jsem si, že takto zrůdně pokračovat nebudu a napsal jsem si program, který dodržuje při řazení českou a slovenskou normu. Vlastnosti programu jsou popsány v článku *Program csr – abecední řazení podle normy.* Článek by měl vyjít v tomto čísle bulletinu.

Nyní rozebereme TEXovská makra, která jsem pro účely obhospodařování databáze členů vytvořil. Především jsem definitivně oddělil databázi individuálních členů a kolektivních, protože údaje, které potřebujeme mít zaneseny, mají v obou případech dosti odlišnou strukturu. Snaha po sjednocení struktury obou databází, která při vzniku počítačové evidence členů existovala, vedla k naprosto neúčelným, nepochopeným a vesměs nevyplňovaným položkám, jako např. položka s názvem meziřádek1 a meziřádek2.

Udržuji tedy dva databázové soubory `individ.tb` a `kolekt.tb`, které obsahují vlastní databázové údaje v textové podobě. Ke každému z těchto souborů je napsán soubor maker `individ.tex` a `kolekt.tex`. TEX se spouští na soubor `*.tex` v němž se pomocí příkazu `\input` přejde na odpovídající databázový soubor. Soubory `*.tex` obsahují tato makra:

- Makra na „scannování", tj. načtení obsahu jednotlivých položek pomocí smluvených oddělovačů.
- Makra na vytváření rešeršních tisků (např. vytiskneme jen členy, kteří mají neprázdnou položku „e-mail" a zároveň nezaplatili za rok

1993)

- Makra na formáty výstupu. V současné době jsou vytvořeny tři formáty: Tisk všech údajů, tisk přehledu plateb (samozřejmě se zabudovaným sčítáním a vyplněním řádku „celkem") a konečně tisk adres na samolepky.
- Za zmínku stojí ještě některá „pomocná" makra. Například makro `\scandatum` přečte datum ve formátu `19940621` a vysází 21. 6. 1994. Makro `\anone` zase vytiskne slovo „ano" nebo „ne" v závislosti na hodnotě položky „logického typu". Nebo makro `\pan` přečte rodné číslo a na základě toho vysází buď „pan" nebo „paní". Omlouvám se všem slečnám; rozdíl mezi „paní" a „slečnou" není z databázových údajů zjistitelný.

Před zpracováním databáze TEXem je potřeba odkomentavat na konci souboru `*.tex` jeden z příkazů `\printall`, `\printmoney` nebo `\printaddress`. Tímto způsobem se stanoví jeden ze tří formátů výstupu. Pokud bychom chtěli vytvořit rešeršní tisk, odkomentujeme navíc jeden z příkazů `\reserse`, nebo vytvoříme nový. Například

```
\reserse{\ifstudent\else\ifx\nezaplatil\moneyTHREE}{\fi\fi}
```

způsobí tisk jen těch členů, kteří nejsou studenty (`\ifstudent\else`) a zároveň nezaplatili za rok 1993 (`\ifx\nezaplatil\moneyTHREE`). Druhý parametr příkazu `\reserse` musí obsahovat odpovídající počet kontrolních sekvencí `\fi`. Vyjadřovací možnosti pomocí `\if` a `\fi` jsou lépe patrné při pohledu do souboru maker. Je totiž potřeba vědět, jak se jednotlivé položky jmenují.

Pokud by bylo potřeba přidat položku nebo změnit strukturu databáze, je nutné pozměnit tu část maker, která se věnuje načítáním položek (tzv. scannovací algoritmy). Pokud by bylo potřeba upravit formát výstupu nebo přidat další typ výstupu, pak se provede jednoduchý zásah do té části maker, která se stará o vzhled výstupu (viz definice tisku). Tyto dvě části maker vzájemně spolupracují, ale je možné je upravovat myšlenkově odděleně.

Uvedená koncepce zpracování databázových údajů se dá použít například při organizování konferencí. Pokud by si to někdo chtěl vyzkoušet pro svou aplikaci, uvádím na závěr pro inspiraci výpis souboru maker `individ.tex`. Aby bylo možno lépe rozumět makrům z tohoto souboru, předchází ukázka jednoho řádku ze souboru `individ.tb`. Každý řádek v souboru `individ.tb` začíná příkazem `\p`. Za ním následují položky týkající se jednoho člena oddělené symbolem : nebo | podle smluvené struktury. Ukázka řádku týkající se jednoho člena se nevejde na jeden řádek v bulletinu. Proto je rozdělena do více řádků. Vězte ale, že ve skutečnosti se jedná o jeden řádkek obsahující zhruba 370 znaků.

```
  \p Ferdinand    Mravenec   :010101/1234
      |Ondřeje Sekory 3  :Praha 13   :11300:     :mravenec@brouk.les.cz
      |Práce všeho druhu :RNDr.   :CSc. :     :                     :
      |     :        :     :     |FFTT: 1993
      |-:     :        : 150:19930709: 150:19940511|
```

Soubor maker individ.tex vypadá takto:

```
% Makro na zpracování databáze členů CSTUGu
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 9.11.1994                      Petr Olšák
% Plain

\newcount\globnum  \newcount\num
\newcount\sumTWO \newcount\sumTHREE \newcount\sumFOUR
\newif\ifstudent \newif\ifvybor \newif\ifrevizor
\newif\ifprihlas \newif\iftisk

% Scannovací algoritmy:
%%%%%%%%%%%%%%%%%%%%%%%

\def\p#1|#2|#3|#4|#5|#6| {\advance\globnum by1
  \begingroup
  \scanname #1|\scanaddr #2|\scanoccup #3|\scanother #4|%
  \scanlogic #5|\scanmoney #6|%
  \expandafter\test\expandafter\print\endtest
  \endgroup}
\def\scanname #1 #2:#3|{%
  \def\jmeno{#1}%                      % jméno
  \def\prijmeni{#2\unskip}%            % příjmení
  \def\rc{#3\unskip}}%                 % rodné číslo
\def\scanaddr #1:#2:#3:#4:#5|{%
  \def\uliceD{#1\unskip}%              % ulice \
  \def\mestoD{#2\unskip}%              % město  -- adresa pro korespondenci
  \def\pscD{#3\unskip}%                % PSČ   /
  \def\telD{#4\unskip}%                % telefon domů
  \def\email{#5\unskip}}%              % email
\def\scanoccup #1:#2:#3:#4:#5:#6|{%
  \def\zamestnani{#1\unskip}%          % zaměstnání
  \def\titul{#2\unskip}%               % titul před jménem
  \def\csc{#3\unskip}%                 % titul za jménem
```

```
  \def\pscZ{#4\unskip}%              % PSČ (zam.)
  \def\uliceZ{#5\unskip}%            % ulice (zam.)
  \def\mestoZ{#6\unskip}}%           % město (zam.)
\def\scanother #1:#2:#3:#4|{%
  \def\telZ{#1\unskip}%              % telefon do zam.
  \def\fax{#2\unskip}%               % fax
  \def\telex{#3\unskip}%             % telex
  \def\poznamka{#4\unskip}}%         % poznámka
\def\scanlogic #1#2#3#4:#5|{%
  \def\jestudent{\anone #1}\if #1T\studenttrue\else\studentfalse\fi
  \def\jevybor{\anone #2}\if #2T\vybortrue\else\vyborfalse\fi
  \def\jeprihlas{\anone #3}\if #3T\prihlastrue\else\prihlasfalse\fi
  \def\jetiskni{\anone #4}\if #4T\tisktrue\else\tiskfalse\fi
  \def\rokclenstvi{#5\unskip}}%     % člen od roku
\def\scanmoney #1:#2:#3:#4:#5:#6:#7|{%
  \def\moneyONE{\anone #1}%         % zaplatil za r. 91 (ano/ne)
  \def\moneyTWO{#2}%                % příspěvek za r. 92
  \def\datumTWO{#3}%                % datum příspěvku 92
  \def\moneyTHREE{#4}%              % příspěvek za r. 93
  \def\datumTHREE{#5}%              % datum příspěvku 93
  \def\moneyFOUR{#6}%               % příspěvek za r. 94
  \def\datumFOUR{#7}}%              % datum příspěvku 94

\def\anone #1{\if#1Tano\else\if#1Fne\else nic\fi\fi}
\let\test=\relax  \let\endtest=\relax
\def\nezaplatil{ 0}
\def\emptyM{ }
\def\emptyP{ \unskip}
\def\reserse#1#2{\def\test{#1}\def\endtest{#2}}
\reserse{\relax}{\relax}
\let\finalaction=\relax

% Definice tisku
%%%%%%%%%%%%%%%

\def\hb{\hfil\break}
\def\scandatum#1{\if#1:\let\next=\eatcolon
   \else\let\next=\makedatum\fi\next#1}
\def\zerox{0}
\def\makedatum#1#2#3#4#5#6#7#8:{%
  \def\tempa{#7}\def\tempb{#5}%
  \ifx\tempa\zerox\def\tempa{}\fi\ifx\tempb\zerox\def\tempb{}\fi
  \tempa#8.\kern.2em \tempb#6.\kern.2em #1#2#3#4}
\def\eatcolon:{}

\def\printall {%
  \let\pt=\sevenrm
```

```
  \hoffset=-1.5cm \advance\hsize by3cm
  \voffset=-1.5cm \advance\vsize by3cm
  \def\beforeprint{\tisktrue}
  \parskip=\smallskipamount
  \baselineskip=10pt \lineskiplimit=-5pt
  \def\print {%
    \global\advance\num by1
    \vbox{\leftskip=\parindent
    \noindent\llap{\bf \the\globnum. }%
    {\bf \prijmeni\ \jmeno}, {\pt RČ:} \rc,
    {\pt Ulice:} \uliceD, {\pt Město:} \mestoD, {\pt PSČ:} \pscD,
    {\pt telD:} \telD.\hb
    {\pt Zam:} \zamestnani, {\pt titul:} \titul, {\pt za jménem:} \csc.
    {\pt email:} {\tt \email},
    {\pt telZ:} \telZ, {\pt fax:} \fax, {\pt telex:} \telex.\hb
    {\pt UliceZ:} \uliceZ, {\pt MěstoZ:} \mestoZ, {\pt pscZ:} \pscZ,
    {\pt Poznámka:} \poznamka.\hb
    {\pt Student:} \jestudent, {\pt Člen výboru:} \jevybor,
    {\pt Přihláška:} \jeprihlas, {\pt Tisknout:} \jetiskni.\hb
    {\pt Od roku:} \rokclenstvi, {\pt příspěvky: }{\pt 91:} \moneyONE,
    {\pt 92:}\moneyTWO\ (\expandafter\scandatum\datumTWO:),
    {\pt 93:}\moneyTHREE\ (\expandafter\scandatum\datumTHREE:),
    {\pt 94:}\moneyFOUR\ (\expandafter\scandatum\datumFOUR:).\par
    }\medskip}
}

\def\l#1#2{\hbox to#1{#2\hss}}
\def\r#1#2{\hbox to#1{\hss#2}}
\def\c#1#2{\hbox to#1{\hss#2\hss}}

\def\printmoney {%
  \advance\hsize by25pt
  \advance\vsize by2\baselineskip
  \def\beforeprint{\tisktrue}
  \headline{\vbox to0pt{\kern-1mm\hbox to\hsize{%
    \hskip3em\l{4cm}{Jméno a příjmení}%
    \l{3em}{členství}\r{3em}{1991}%
    \c{9em}{1992}\c{9em}{1993}\c{9em}{1994}\hss
    }\kern2pt\hrule\vss}}
  \def\print {%
    \global\advance\num by1
    \ifx\moneyTWO\emptyM\else\global\advance\sumTWO by\moneyTWO\fi
    \ifx\moneyTHREE\emptyM\else\global\advance\sumTHREE by\moneyTHREE\fi
    \ifx\moneyFOUR\emptyM\else\global\advance\sumFOUR by\moneyFOUR\fi
    \noindent
    \r{3em}{\the\num.\ \ }\l{4cm}{\prijmeni\ \jmeno}%
    \l{3em}{\rokclenstvi\ifstudent*\fi}%
```

```
    \r{3em}{\moneyONE}%
    \r{3em}{\moneyTWO}\r{6em}{\expandafter\scandatum\datumTWO:}%
    \r{3em}{\moneyTHREE}\r{6em}{\expandafter\scandatum\datumTHREE:}%
    \r{3em}{\moneyFOUR}\r{6em}{\expandafter\scandatum\datumFOUR:}%
    \par
    }
  \def\finalaction {%
    \medskip\hrule\medskip
    \noindent\hskip3em
    \l{4cm}{\bf Celkem}\hskip6em\r{3em}{\the\sumTWO}\hskip6em
    \r{3em}{\the\sumTHREE}\hskip6em\r{3em}{\the\sumFOUR}\par
    }
}

\def\pan{\expandafter\scanrc\rc}
\def\scanrc#1{\if#1\unskip pan\let\next=\relax
  \else\let\next=\scanrcnum\fi \next#1}
\def\scanrcnum#1#2#3#4\unskip{\ifnum#3>4 paní\else pan\fi}

\def\mkpsc{\expandafter\scanpsc\pscD}
\def\scanpsc#1{\if#1\unskip\let\next=\relax\else\let\next=\scanpscnum\fi
  \next#1}
\def\scanpscnum#1#2#3#4\unskip{#1#2#3\thinspace#4\unskip}

\def\printaddress{
  \baselineskip=1.1\baselineskip
  \nopagenumbers
  \hoffset=-1in  \hsize=210mm
  \voffset=-1in  \vsize=297mm \raggedbottom
  \lineskip=0pt
  \def\print {%
     \iftisk
     \global\advance\num by1
     \ifnum\num=16\par\global\num=1\fi
     \noindent\vbox to 37mm{\hsize=70mm \leftskip=18mm\vss
     \noindent \pan \hb
     \jmeno\ \expandafter\uppercase\expandafter{\prijmeni}\hb
     \hbox{}\uliceD\par
     \noindent\llap{\mkpsc\enspace}{\bf\mestoD}
     \vss}\hskip0pt plus2pt
     \fi}
}

% Vlastní zpracování (rešerše a deklarace formátu tisku)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Rešerše se zadávají pomocí příkazu \reserse se dvěma parametry.
```

```
% První parametr obsahuje vlastní podmínku a druhý parametr odpovídající
% počet \fi-ů.
% Je možno použít: \ifstudent \ifvybor \ifrevizor \ifprihlas \iftisk.
% Testy na položky typu \ifx, přičemž prázdá položka se testuje na
% \emptyP, prázdná položka v údaji o zaplacení na \emptyM a nulová
% hodnota v údaji o zaplacení má reprezentaci \nezaplatil.
% Následující příklad vybere všechny členy výboru, kteří mají neprázdnou
% položku email:

% \reserse{\ifvybor\ifx\email\emptyP\else}{\fi\fi}

% Jiný příklad vybere všechny členy, kteří nezaplatili za rok 93:

% \reserse{\ifx\nezaplatil\moneyTHREE}{\fi}

% Další případy:
% \reserse{\ifnum\num>30\endgroup\end\else}{\fi}
% \reserse{\ifstudent}{\fi}

% Fomát tisku se vybere z následujících příkazů. Lze definovat
% analogicky další formáty (viz část definice tisku).

%\printall     % vytiskne všechny údaje v dosti zhuštěném tvaru.
\printmoney   % vytiskne prehled plateb.
%\printaddress % vytiskne adresy na lejblíky

\input individ.tb
\finalaction
\end
```

---

8. 12. 1994 *Petr Olšák*